

N-ary data in Semantic MediaWiki

Yaron Koren

Spring 2010 SMWCon
May 23, 2010

What is n-ary data?

Any information that cannot be expressed as
simple triples

N-ary data is all around us!

Simple statement:

"Alice bought shoes"

No problem!

Alice bought shoes

Slightly more complex statement:

"Alice bought shoes for \$50 at The Gap"

???

This is an n-ary relation!

How can it be expressed as triples?

Answer 1:

We use an intermediate, "blank node".

Alice made purchase X
X has item shoes
X has cost \$70
X has store The Gap

Answer 2:

We set all the values as the "object" of the triple.

Alice made purchase (shoes, \$70, The Gap)

There are currently 3 1/2 solutions for using n-ary data in Semantic MediaWiki:

- Semantic Internal Objects extension
- "Record" type in SMW
- External Data extension
- SemanticHistory extension

Solution 1: Semantic Internal Objects

Created by Yaron Koren, first released August 2009

Defines "internal objects" within a wiki page that mimic blank nodes

Storage is done via a parser function,
`#set_internal`.

Semantic Internal Objects - usage

On the page called "Alice":

```
{ {#set_internal:Is purchase of|Has  
item=shoes|Has cost=$70|Has store=The  
Gap} }
```

All four of these are stored as properties of a quasi-wiki page, "Alice#1".

"Is purchase of" points back to main page.

Why "Is purchase of" and not "Has purchase"?

To enable easy querying.

To get all of Alice's purchases:

```
{ {#ask: [[Is purchase of::Alice]] |  
mainlabel=- | ? Has item=Item | ? Has  
cost=Cost | ? Has store=Store} }
```

To get everyone's purchases:

```
{ {#ask: [[Is purchase of::+]] |  
mainlabel=- | ? Is purchase of=Buyer | ?  
Has item=Item | ? Has cost=Cost | ? Has  
store=Store} }
```

Solution 2: “Record” type

First added to SMW in version 1.0 (December 2007) as "multi-value properties"; re-implemented as the "Record" type in version 1.5 (March 2010).

Sets all values as the object of the triple.

In page "Property:Made purchase":

```
[ [Has type::Record] ]
```

```
[ [Has fields::String; String;  
    String] ]
```

In page "Alice":

```
[ [Made purchase::Shoes; $70; The  
    Gap] ]
```

Solution 3: External Data

Created by Yaron Koren; first released January 2009.

Improved by many people.

Non-SMW-based solution for n-ary data: store a table of data in a CSV file, database, etc., then query it using External Data's parser functions.

Solution 3.5: SemanticHistory extension

Created by Jie Bao and Li Ding; first released
September 2009.

Semantically stores all previous revisions to any
page.

SemanticHistory

Only useful for a very specific kind of n-ary data:

subject **relationship** **object** **at time t**

Assumes that the information in the wiki is correct
at any given time.

In my opinion:

Semantic Internal Objects is the best solution.

Advantages of Semantic Internal Objects

- It's actually semantic; every value is associated with a named property
- Supports enumerated properties
- Flexible; additional fields can be added easily
- Querying is more straightforward

Disadvantages of Semantic Internal Objects

- It's a hack! SMW was not designed with SIO in mind. Display of internal objects is awkward for that reason.
- Has a bug that sometimes leads to extra internal objects when certain pages are re-saved. Hopefully this will be fixed soon!
- Data is not exported as RDF

Editing n-ary data

For both Records and SIO, data should ideally be stored using multiple-instance templates

Possible better interface: editable grid

This may be a component of Sanyam Goyal's upcoming Google Summer of Code project, which will convert Semantic MediaWiki and some related extensions to use jQuery.

Import of n-ary data

No good solution exists for this! Data Transfer does not currently support multiple-instance templates.

External Data supports n-ary data, but this is "use"/"query" of data, not import.

Export of n-ary data

No RDF export yet for any of the solutions.

Data Transfer's XML export supports multiple-instance templates.